

APPLICATION FOR UNITED STATES LETTERS PATENT

WORKFLOW DRIVEN RULES-BASED GENERATION  
OF PERSONALIZABLE WEB PAGES

Inventors:

Murugan Palaniappan  
20045 Glen Brae Drive  
Saratoga, CA 95070  
A Citizen of India

Robert Brian Schmidt  
284 Everett Avenue  
Palo Alto, CA 94301  
Citizenship: USA

Mark Dao  
151 South Bernardo Ave. #40  
Sunnyvale, CA 94086  
Citizenship: USA

Assignee:

Asera, Inc.  
600 Clipper Drive, Suite 100  
Belmont, CA 94002  
A Delaware corporation

Beyer, Weaver & Thomas LLP  
P.O. Box 778  
Berkeley, CA 94704  
Tel: (650) 961-8300

Atty Dkt No. ASRAP013

WORKFLOW DRIVEN RULES-BASED GENERATION  
OF PERSONALIZABLE WEB PAGES

5

## RELATION TO PRIOR APPLICATIONS

This application is related to the following -- U.S. Provisional patent application having serial number 60/164,021 (Attorney Ref. No. ASRAP001P), entitled "Method and Apparatus to Provide Custom Configurable Business Applications From a Standardized Set of Components," filed August 23, 1999; Utility patent application having serial number 09/440,326 (Attorney Ref. No. ASRAP001), entitled "Method for Providing Custom Configurable Business Applications from a Standardized Set of Components," filed November 15, 1999; Utility patent application having serial number 09/439,764 (Attorney Ref. No. ASRAP002), entitled "Apparatus to Provide Custom Configurable Business Applications from a Standardized Set of Components," filed November 15, 1999; Utility patent application having serial no. 09/658,415 (Attorney Ref. No. ASRAP003), entitled "Method For Developing Custom Configurable Business Applications," filed September 8, 2000; Utility patent application having serial no. 09/658,416 (Attorney Ref. No. ASRAP014), entitled "Integrated Design Environment For A Commerce Server System," filed September 8, 2000; Utility patent application having serial no. 09/697,271 (Attorney Ref. No. ASRAP005), entitled "Method for Providing Template Applications For Use By a Plurality of Modules," filed October 25, 2000; Utility patent application having serial no. 09/691,461 (Attorney Ref. No. ASRAP006), entitled "Method and Apparatus for Providing News Client and Server Architecture and Protocols," filed October 17, 2000; Utility patent application having serial no. 09/684,491 (Attorney Ref. No. ASRAP008), entitled "Adapter and Connector Framework for Commerce Server System," filed October 4, 2000; Utility patent application having serial no. 09/702,148 (Attorney Ref. No. ASRAP009), entitled "E-Commerce Application Built Using Workflows on a Workflow Engine and Methods Thereof," filed October 30, 2000; Utility patent application having serial no. 09/702,290 (Attorney Ref. No. ASRAP010), entitled "Presentation Layer For Business Application Development And Methods Thereof," filed October 30, 2000; Utility patent application having serial no. 09/702,291 (Attorney Ref. No. ASRAP012), entitled "Scalability, Availability, and Management Features For Business Commerce Server," filed October 30, 2000; and Utility patent application having serial no. 09/706,304 (Attorney Ref. No. ASRAP011), entitled "Content Management Framework For

Business Commerce Server," filed November 3, 2000 -- each of which are hereby incorporated by reference in their entirety.

## FIELD OF THE INVENTION

5 The present invention relates generally to techniques for layout of personalizable web pages. The personalizable portion of a web page generally contains condensed views of business applications' data and functionality. The presentation of such a page can be determined according to certain dynamically evaluated rules, which govern both the overall page layout and style as well as the configuration of individual components of the layout  
10 according to the rule conditions.

## BACKGROUND OF THE INVENTION

The prior referenced applications provide for methods and apparatuses for creating custom configurable business or channel applications from a standardized set of components. More specifically, the referenced invention allows each business to select from a set of 15 applications, customize that set of applications, and/or develop new customized applications from a set of development components. The prior applications provide for a server based method wherein best-of-breed services and/or applications are integrated in a seamless fashion and offered to enterprise businesses which develop customized business service applications through using the system. The server device is previously (and hereafter) referred to as the  
20 Asera Commerce Server (ACS).

The ACS includes a Commerce Server that provides a core set of technology (or application) services. A unique architecture and framework are provided by the Commerce Server, which facilitates development and use of customized applications. Most interactions with external systems or users are managed as Business Objects. The service application code 25 is maintained separate from the data. This enables the system to quickly include (and/or change) new business processes or technology components without having to write substantial amounts of new code. The business result is more rapid customer deployments and/or modifications that are customized to include (if desired) the proprietary or competitive business practices of a contracting company.

The ACS can be viewed as a form of ASP (Application Service Provider). An ASP is generally an outsourcing business model. The ASP business model requires an open and extendable architecture that allows a system to implement a customer specific business solution in a short period of time. The ACS takes best-of-breed applications and incorporates them into one integrated solution to provide the ASPs. The architecture is scalable and extensible. A customized business (or channel) application solution is built for each enterprise company. The solution uses a "modular" or step-wise or "plug and play" approach towards building new applications. An enterprise company can then quickly acquire a turn-key e-commerce solution to automate their channel relationships. The system presents little (or no) risk for the enterprise company because a solution is built by the present system. The costs of undertaking such a development exist as a fixed development cost of the system. Any resulting customized solutions are implemented in considerably less time than previous systems. The enterprise company might pay for the application services on a cost per transaction, or a fixed fee basis.

The ACS is used to capture the particularized (or specific) business processes for a given customer, and these business processes are converted into a set of customized applications. The ACS uses business steps and rules to construct the application. The objects are data representations. The steps are business operations with a defined set of input and output ports, with each port also having a defined set of parameters. The business rules are used to capture customer specific business practices. A unique tool that employs a graphical user interface (GUI), allows a developer to arrange various steps (or composite steps) into business processes, or workflows. The tool provides library catalogs of steps to be applied to the various objects. The connections between steps are also verified as correct. A graphical display of the business process is shown, and rules can thereafter be applied to provide further customization by conditionally tagging certain points. Hence, to create a business process (or application) for any given business, tools are provided which allow modules (or steps) to be plugged or dropped into the potential process. The steps can be moved, or the connections modified. An initial person-to-person (or other type of) interview with the business (or customer) can be used to produce the framework for arranging the steps according to the needs of that particular business (i.e. customized routines). The modular aspect of the present system allows this to be done -- and modifications made -- in a relatively quick fashion. For instance, if a process has been created, but the customer wants it to behave in two different

manners, then certain rules can be applied to provide the desired results, depending on conditional triggers that can be associated with the underlying Business Objects.

In general, Internet transactions can be divided into two categories: 1) business-to-business transactions, and 2) business-to-consumer transactions. Most solutions to automate transactions have dealt with business-to-consumer interactions. As such, these interactions are much more straightforward than business-to-business transactions. In a business-to-consumer transaction, the merchant supplies a "storefront" or web site that offers products to any number of diversified consumers who might wish to view this web site. The consumer then purchases a product via a selection and payment method, and the product is thereafter shipped to the consumer. On the other hand, when one business deals with another business, there is a much greater amount of business processes and customization in the transactions that occur. The ACS system therefore provides a customized business application that runs on the server and communicates with outside data sources.

Prior examples of such commerce systems have provided a portal page that serves as a convenient area for a user to collectively view different types of information. Portal pages have been generally limited, however, in their level of integration with the underlying applications whose data is presented on the portal page. In particular, prior portal pages have lacked the advantages in design, implementation, and execution that come from tight integration at the level of workflows.

Portal pages have also been limited in their ability to display application data according to user specified conditions, or rules. A user in prior portal pages can generally arrange the layout and customize the content of the portal page according to their preferences. However, once these preferences are established, the portal page will be displayed in the same way until the user explicitly changes his or her preferences; in other words the behavior is static.

Accordingly, what is needed in the field is the ability to provide a portal page that can display and simultaneously run a variety of workflow applications. Additionally, the portal page might employ a rule device that displays the contents of the portal page according to conditions that satisfy the rules, thus fundamentally changing the nature of the portal page by allowing for multiple sets of user and administrator preferences.

## SUMMARY OF THE INVENTION

The present invention implements a dynamic wireframe extension to the ACS Presentation Layer that facilitates development of personalizable web pages. Furthermore, the design, the implementation, and the execution of Portal Objects on a portal page is integrated 5 within the larger business applications for which the Portal Objects are acting as personalizable windows, or views.

Additionally, web pages rendered using the dynamic wireframe mechanism of the present invention reflect both the explicit preferences of the end user and the static and dynamic evaluation of rules governing specific aspects of the page. Users can explicitly make 10 style selections, modify the page layout, and enter preferences for individual content items, or Portal Objects. This is similar to the core capabilities of other portal products. In the present invention, however, a hierarchy of rules can be established in order to make the realization of various aspects and components of the portal page a conditional determination. In prior products, a user can set preferences for displaying content on a web page. However, there is 15 no facility for creating multiple sets of such preferences, and for conditionally choosing from among those preference sets based on the evaluation of rules.

According to one aspect of the present invention, Portal Objects shown on a portal page can be developed using workflow as part of the application they represent, thereby reducing the cost and difficulty of development and increasing the integrity of the Portal Object with 20 respect to issues such as UI consistency, globalization, entitlement, and security that are governed by the underlying ACS system.

According to another aspect of the present invention, rules can be created so that a single logical location, or page, within a web site can have multiple dynamic underlying "physical" 25 manifestations even for a single user. Furthermore, a single logical Portal Object instance on a physical page manifestation can have multiple dynamic underlying physical Portal Object manifestations that depend upon the execution of additional rules.

These and other aspects and advantages of the present invention will become apparent upon reading the following detailed descriptions and studying the various figures of the drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

5 The invention, together with further advantages thereof, may best be understood by reference to the following description taken in conjunction with the accompanying drawings in which:

Figure 1 shows, according to one aspect of the present invention, a representative page layout structure with columns of fixed width.

10 Figure 2A shows, according to one aspect of the present invention, a two-column layout with section demarcations.

Figure 2B shows, according to one aspect of the present invention, a column with locked and user chosen areas.

15 Figure 3 shows, according to one aspect of the present invention, the correlation of the presentation of Portal Objects on a portal page with underlying application workflow.

Figure 4 shows, according to one aspect of the present invention, a hierarchy of properties as applied to a Portal Object parameter such as number of news headlines.

Figure 5 shows, according to one aspect of the present invention, an example of rule-based selection of portal page style merged with user style preferences.

20 Figure 6 shows, according to one aspect of the present invention, a prior art example of web pages related to Portal Objects.

Figure 7A shows, according to one aspect of the present invention, an example of a Rule Definition Wizard being applied to portal page layout preferences.

25 Figure 7B shows, according to one aspect of the present invention, an example of the Rule Definition Wizard layers.

Figure 8A shows, according to one aspect of the present invention, an example of rule-based selection of a dynamic wireframe layout.

Figure 8B shows, according to one aspect of the present invention, an example of rule-based selection of Portal Object properties.

5

## DETAILED DESCRIPTION OF THE INVENTION

The present invention is described in terms of examples relating to the above-referenced ACS system and relies on many of the concepts and principles as the prior ACS Presentation Layer and Infrastructure, as described in the set of incorporated references. The present invention is not, however, intended to be limited to usage on this particular system, and is 10 broadly applicable to other types of server and networking devices.

### PORAL PAGES

The benefits of a web site built for facilitating commerce transactions are often only fully attainable if individual users can improve, or optimize, their experience of the tasks they accomplish within the site. One approach to this problem is to allow users to control, within 15 certain limits established by administrators, the content, the layout, and the style of selected web pages. After a user personalizes a page to match his or her needs and preferences, that page, as well as the larger web site to which the page belongs, can be utilized more efficiently and effectively by that user. And, partly as a byproduct of the requisite investment of time, these personalizable web pages can serve as a retention mechanism that strengthens a user's 20 loyalty to the site.

A personalizable web page generally has one or more of the following three defining characteristics:

1. Users are permitted to personalize the arrangement of content on the page, often by selecting from among named "content items" representing data drawn from various 25 disparate applications or other sources of content irrespective of the protocols.
2. Users are permitted to personalize the rendering of the information *within* the various discrete and independent content items displayed on the page.

3. Users are permitted to personalize the style, or presentation aspects, of the page.

Pages with such capabilities are often labeled "portal" pages because, as aggregators of content from disparate sources, they can simultaneously serve as (a) an effective entry point into a web site, or a portion of a web site (b) a personal workspace or "sandbox" for end users, 5 and (c) an anchor, or "home" page within the web site, or a portion of a web site, to which the user can quickly navigate, even from deep within a complex hierarchy of web pages.

The present invention extends the ACS Presentation Layer by providing a "dynamic wireframe" construct that allows for rapid and easy development of personalizable web pages. Creation of portal pages is a primary usage of the dynamic wireframe, but the construct is 10 generally applicable to any or all web pages rendered using the ACS Presentation Layer where an application developer can develop new applications using the portal page infrastructure.

## PORTRAL OBJECTS

The present invention also formalizes the notion of "content items" to be used within personalizable web pages, hereafter called "Portal Objects", and establishes guidelines for 15 developing Portal Objects using the ACS Integrated Design Environment. A Portal Object is developed using the same workflow, filter, criteria, candidate, entitlement, etc. constructs and methods that are used for regular application development. Usually, a Portal Object is written to provide a condensed view of the data and functionality of a particular business application (although there are no limitations on the cardinality of relationships between applications and 20 Portal Objects). In this case, the workflows, the business logic, filters, criteria, candidates, etc. that constitute the Portal Object can be shared, and re-used, with the application(s) associated with the Portal Object (and vice versa). In particular, the Portal Object's functional and interaction workflows can be written and integrated within the workflows of the larger application. Portal Objects can also be written as "standalone" pieces of workflow that are not 25 associated with a larger application.

When the user exercises one of the Portal Objects, e.g. by using the search form for the catalog application, then the user is simply continuing along a well-defined path within that application's workflow. The portal page itself, therefore, is a composite of multiple pieces of workflow that are dynamically stitched together so that the user can simultaneously peek into 30 a variety of applications.

A Portal Object's "behaviors" are specified as top-level interaction steps in the workflow. Every Portal Object will identify such a step, with a corresponding port, as the implementation of the "display" behavior. This workflow is responsible for rendering the Portal Object. Portal Objects may also specify additional steps that implement "personalize" and "configure" behaviors that will, correspondingly, allow users and administrators to enter preferences that affect the rendering of the Portal Object. The Portal Object "behavior" specification is a general one that can be extended to include additional behaviors as the need arises. During the deployment of the ACS server, Portal Objects can be registered for use on personalizable pages. A Portal Object's behaviors can then be elicited by executing the workflow that is associated with that behavior.

The principles, concepts, and components described and implemented in the ACS Presentation Layer apply in full to the development of Portal Objects. In particular, Master templates, Wire Frames, Micro Templates, Behavior Filters and Display Filters are used to declaratively specify the UI of a Portal Object.

As per the example portal page 300 shown in Figure 3, application examples include: (a) Top News Headlines 302; (b) a search entry point into a product catalog 304; (c) a Message Board 306; and (d) user outstanding Auction Bids (with status), and associated links to bid details 308. These applications are represented on a portal page by the respective Portal Object workflows including: News Portal Object 312, Catalog Portal Object 314, Message Board Portal Object 316, and Auction Bid Portal Object 318.

Note that other portals do not generally allow for a workflow (as used in the ACS system) to be tied into the different portal areas. For example, if a user goes to his or her portal page on Yahoo and proceeds to click on an item, the user will be taken to a different page, or a different set of applications (which often have a different format). In the present invention, if a user enters his or her portal page and clicks on a displayed application, then the user will continue to exist within the same overall shell. In particular, if the user clicks on a Portal Object, then the workflow can continue using an Output Port of the Interactive Step which was responsible for rendering the Portal Object. Each window within the portal page operates within its own workflow. Accordingly, because of the inherent reliance on workflow, previously described concepts (in the referenced applications) such as Criteria, Candidates, Display Filters, Behavior Filters, View Display Objects (VDO), Entitlement, and

so forth are already bundled with and available during the development and the execution of Portal Objects.

Figure 6 shows an example of how the user interacts with a Portal Object known in the prior art. The Portal Object 604 is rendered on a portal page to include a news area 610, 5 which might include a list of top news headlines. An edit, or personalize, button 612 invokes a Portal Object Preferences page 606, wherein the user can set preferences, for example, for the source content of the news headlines. If the user clicks on one of the headlines shown on the portal page, then a News Application page 608 is displayed that contains the headline content.

## DYNAMIC WIREFRAME

10 The present invention provides for dynamic rendering of a web page wherein each slot of a dynamic wireframe contains a view of an application that is running on the commerce server system (i.e., ACS system). Each application can define multiple views, or Portal Objects, that are appropriate for display on the portal page. A portal page requires a special implementation of an Interactive step that will serve to arrange items on the page but will 15 delegate the job of actually rendering the content of the page to independent pieces of workflow. This is achieved by embedding a dynamic wireframe reference within the static wireframe of the Interactive Step.

The mechanism by which the portal page is rendered involves special usage of interactive steps as described in association with the ACS Presentation Layer. Since the 20 layout of a portal page may be different for every user of the system, the normal static wireframe mechanism is not sufficient. The interactive step that renders the portal page must instead act as a "dynamic wireframe" in that the wireframe contents and layout are created on the fly for the current user. Each Portal Object in a dynamic wireframe instance is equivalent to a single Micro Template Tag in a static wireframe. Furthermore, the execution of the 25 dynamic wireframe requires workflow from the various Portal Object to be dynamically stitched together in order to render the Portal Page. Thus, the dynamic wireframe acts as "glue" for independent pieces of workflow that collectively produce the contents of a personalized portal page.

When the Interactive Step for a portal page is processed, the dynamic wireframe will spawn off requests for content from each of the Portal Objects it contains. Accordingly, workflow for each of the applications to which the Portal Objects belong will be executed. The result will be HTML snippets that will be rendered in the appropriate location within the dynamic wireframe. Thus, when a portal page is encountered, the execution of the dynamic wireframe is going to simultaneously execute workflow for the different applications corresponding to the various Portal Objects arranged within the dynamic wireframe. Each of these Portal Objects, in turn, will have its own workflow that will terminate in an Interactive Step that renders the HTML for the Portal Object using the regular wireframe, filter, entitlement, etc. mechanisms. Portal Objects that the user is not entitled to view can be automatically removed from the layout before the rendering takes place.

A dynamic wireframe is essentially a two dimensional arrangement of Portal Objects into columns of fixed width, each column containing any number of individual Portal Objects stacked vertically. Referring to Figure 1, a dynamic wireframe layout 100 usually consists of a main (wide) column 102, with a narrow column on the left side (Left Column 104), the right side (Right Column 106), or both sides (as shown). There is, however, no restriction on the absolute or relative widths of the columns in a layout.

Each column is divided into top, middle, and bottom sections. At the top are generally Portal Objects that have been selected, locked, and ordered by the administrator. In the middle are Portal Objects that have been selected and ordered by the end user. At the bottom are more Portal Objects that have been selected, locked, and ordered by the administrator. Locked Portal Objects cannot be reordered and cannot be removed by the end user. Figure 2A shows a generalized example 200, with a first column 202 and a second column 204. Column 1 shows a top area 206 and a bottom area 208 where the user cannot change the layout. A middle area 210 is shown where the user can change the layout. Figure 2B shows a more particularized example of a column 220. The top area includes an example Locked Portal Object A 222 and B 224. The middle area includes example User Chosen Portal Objects C 226, D 228, and E 230. The bottom area includes an example Locked Portal Object F 232.

Note that splitting a column into sections for design purposes does not imply that the Portal Objects will appear to the user in "chunks." The Portal Objects in the column will be rendered in a consistent manner from top to bottom. The purpose for defining different

sections is so that certain Portal Objects can be locked by administrative users in accordance with the design objectives of the portal page.

A page rendered using the dynamic wireframe method also affords the user the ability to personalize the layout of the page. As indicated above, end users can add Portal Objects to, 5 remove Portal Objects from, and move Portal Objects up and down within, the middle section of each column of a dynamic wireframe layout. Administrative users can add/remove/move all Portal Objects in a column. This is generally considered "standard" portal page functionality.

#### RULE BASED LAYOUT ASSEMBLY

10 The present invention provides for hierarchical rule based selection of a layout used for the dynamic wireframe of a personalizable web page. A Rules Definition Wizard (or the like) is provided for both administrators and end users to specify the conditions associated with one or more candidate layouts for a given dynamic wireframe.

15 At the first level, administrators can create multiple default candidate layouts for a dynamic wireframe and associate a condition with each. For instance, an administrator for a company with three different user constituencies, Sales, Product Development, and General can create three different layouts for a portal page. Each layout can contain common Portal Objects plus Portal Objects that are specific to that user constituency. A rule can then be created to choose a dynamic wireframe layout based upon a user's job or department.

20 At the second level, end users can also create multiple candidate layouts for a dynamic wireframe, which are all derived from the appropriate first-level default layout. As an example, a user in Sales might be working from home part of the time, and working from the office the other part of the time. It may be desirable to maintain two versions of the portal page if the content to be viewed at these different locations differs by category, verboseness, 25 or presentation. For office use, the user might want the portal page to display full details of news regarding prospective business deals (or the like) and other similar content. For home use, the user might want the portal page to display the user's personal stock portfolio (or the like), relevant condensed headlines, and other similar content. Referring now to Figure 8A, if logging in from Home 802, then a certain dynamic wireframe Layout 806 will be rendered. If

logging in from the Office 804, then a different Layout 808 will be rendered. Other conditions can act as proxies, whereby, for instance, the user's home display preferences will automatically be effective after 8 pm, and the office display preferences will be effective before 8 pm.

5        The same rules-based selection method can be applied at the Portal Object level. In one scenario, a rule can govern the visibility of a Portal Object so that, for example, a "News Headlines" Portal Object is rendered only if there is at least 1 headline less than 12 hours old in one of the headline categories chosen by the user. As another example, an administrator could arrange for a "Promotion" Portal Object to appear on the page only when a user's total  
10 spending on the web site crosses a certain threshold.

Finally, the configuration of individual Portal Objects can vary according to rules. In this way, for instance, a Promotion can be configured to show material that is targeted to the type of user that is viewing it. As well, an end user can explicitly create two sets of preferences for the same Portal Object, one of which is used when he or she is accessing the site from home and the other which is to be used while at the office. Referring now to Figure 8B, the preferences used when logging in from Home 820 will cause Portal Object instance 824 to be rendered, which shows 2 headlines from each of 2 news sources. Likewise, the preferences used when logging in from the Office 822 will cause Portal Object instance 826 to be rendered, which shows 4 headlines from each of 3 news sources.

20        Additionally a hierarchy can be used for determining what particular properties are to be associated with the Portal Objects on a portal page. Referring to Figure 4, a hierarchy 400 is demonstrated involving entitlement or access levels, such as user, administrator, or the like. As a further part of the example, the number of news headlines to be displayed in the News window of the portal page is the property 408 to be determined. Accordingly, the process will  
25 progress through the access levels, with the highest level of the property (in terms of personalization) being used first. For the number of headlines, the user at the user level 402 might have set a desired number of headlines to be 5 after 8pm and 10 before 8pm. If, however, the user has not yet personalized this Portal Object, then the administrator at level 404 might have set the headline number to be 3. In other words, when the administrator  
30 created the page to have the news portal object on the page, the administrator might have set the number of headlines to be 3, and this number will be used if the user has not set a different

number. If neither the user nor the administrator has set a headline number, then a default level 406 might be used, wherein the number of headlines might be set to 2. Accordingly, when the news headline window is rendered, the appropriate number of headlines will be displayed, depending upon the determined value of this property.

5        In summary, the present invention provides a solution that allows for the content of a personalizable web page to be determined dynamically based upon the evaluation of rules. The various components of a personalizable page can be made, using rules, to depend upon attributes of the current user as well as miscellaneous environment variables. Several example manifestations of this functionality include (but are not limited to):

10      (a) A layout for a portal page can be conditionally chosen based upon the time of day, e.g., use layout A if before 7 pm, otherwise use layout B.

      (b) The visibility of a Portal Object can be conditional, e.g., only show this special promotion if the user has attained a certain spending level or belongs to a specific group.

15      (c) The content shown by a Portal Object can be conditional, e.g., show an advertisement from group A if the user enjoys sports, or from group B if the user has literary interests.

      A further distinction is that some of these rules can be created explicitly by a user, e.g., the user might want a different layout depending on the time of day. Still other rules can act implicitly from the user's point of view. Such implicit rules are defined by an administrator  
20 and act without the user's knowledge.

      Referring now to Figure 7A, a block diagram 700 is shown of representative elements associated with the rule-based generation. In this embodiment of preferences 702, the user will be provided with certain choices 704 to be displayed on the portal page. The user will also be presented with a Rules Definition Wizard 706 (or the like) for establishing conditions  
25 and candidates. For example, an attribute or condition that might be tested by the rules is whether the user is logging in from Home (708) or from the Office (710). Whereas prior methods simply establish a single set of preferences, this rule-based system provides for a programmatic evaluation during runtime of conditions that will cause different sets of preferences to be chosen.

The Rules Definition Wizard 706 might provide for rule definition or generation in any of a variety of ways. Figure 7B shows an example of the Rules Definition Wizard 706 including an instance of Business Rules Layer 730 and an Interface Layer 732. The Interface Layer 732 is generally used to represent an interface between the User 734 and the Business Rules Layer 730. Java programming might be used to implement specific rules into the Business Rules Layer 730. In particular, the system might ultimately use Java snippets for such rules. The User 734 however might expect to see an English based interpreter in order to easily formulate rules. Many other types of Interface Layers 732 might be used, including menus, drop-down lists, and the like. The key is to provide an additional semantic layer in order to take the various forms of user input and properly convert such input into the Java snippets (or the like) to be used by the Business Rules Layer.

The rules will conditionally act on attributes associated with and exposed by the Portal Objects on the portal page. Prior systems found it necessary to determine and hard-code certain Business Object attributes into the conditionality of the rules. However, note that under the present system, a Business Object that conforms to the present standards of the ACS system will have a known set of attributes that can easily be exposed dynamically by the Rules Definition Wizard in order to provide conditional testing according to those attributes. Accordingly, if a new Business Object is created in the ACS system, rules can be readily applied via exposure of the Business Object attributes.

## 20 STYLE

Each user can also personalize certain aspects of the style of a portal page. This can include, but is not limited to, colors and fonts. Style choices can be made at the level of individual attributes, e.g. Portal Object background color, or the user can be offered a selection of "style sets" that contain values for all style attributes which collectively make good visual sense.

These user-chosen style attribute values, if any, are overlaid upon the default style of a portal page. This default style is also determined using rule evaluation, so that different styles can be presented to different sets of users. This is especially important when the entire web site is presented, or "branded", differently for different groups of users. For instance, a single web site may need to present similar content and functionality to users belonging to two

different Vendor organizations. Users from Vendor A should see pages that are branded with the logo of Vendor A, and in which the color red is prominent. Likewise, users from Vendor B should see pages that are branded with the logo of Vendor B, and in which the color blue is prominent. This is achieved by creating two default page styles, and choosing between them 5 based on a rule that evaluates the organization of the current user. This rule-based style mechanism will also encompass Master Template selection, CSS selection, etc., as discussed in referenced inventions, for the purpose of ensuring that the look and feel of the site is consistent across all pages, not just the "portal" pages.

Figure 5 shows an example in which for Vendor A 502, the default style values 506 are 10 "red color, Times 10pt font" and for Vendor B 504, the default style values 508 are "blue color, Tahoma 11pt font". When a user accesses the portal page, then his or her personal style preferences 510, "12pt font" will override the relevant default value. Thus, if the user belongs to Vendor A, then the resulting portal page style attributes 512 are "red color, Times 12pt font", whereas if the user belongs to Vendor B then the resulting portal page style attributes 15 514 are "blue color, Tahoma 12pt font". The user's preference for fonts to be displayed in 12pt takes precedence over the value specified in the relevant vendor-specific default style.

Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced within the scope of the appended claims. Therefore, the described embodiments 20 should be taken as illustrative and not restrictive, and the invention should not be limited to the details given herein but should be defined by the following claims and their full scope of equivalents.